Ye Zee News

Issue # 2
August 1984

Well, here is the next issue and as you can see we now have a name. Some of the things scheduled for this issue have not been included, some because they are not ready. Others just did not fit, at $1 a copy I can not use more than 3 pages per issue until Ve Zee's Post Office registration comes through (soon I hope) thus allowing cheaper postage.

The response with questions and contributions has been good so far, so keep it up.

Cheers,

John

QUESTIONS and REQUESTS

PRINTERS

A few readers have asked questions about output of inverse and graphics to printers. I have a TP-40 printer/plotter and inverse/graphics characters can not be printed using COPY, LPRINT or LLIST. The reason for this is that the printer accepts ASCII codes for the character to be printed, these characters are stored in a ROM inside the printer. If you switch the printer on while you hold down the 'paper feed' switch the full character set will be printed. In the text mode these are the only printable characters.

If you have the time and patience a program could be written to use the graphics mode to print inverse/graphics (in only 4 colours) however I think it would be a big job.

According to the BASIC reference manual inverse/graphics can only be printed on a 'SEIKOSHA GP-100 or GP-100A printer. One reader has tried a Tandy CGP-115 without any success (except with normal text).

The reverse line feed LPRINT CHR$(11) does not work on my TP-40 (or the one at the local Dick Smiths). In fact it results in a wasteful ejection of paper. Does anyone know how to get it to work.

Can anyone help with the following

How do you make CLOAD'ed programs auto start?
How do you merge two BASIC programs from tape?
How do you make Machine Code Program tapes?

Wanted

Circuit diagram for joystick interface.
Information on Disk drives, disk controlers and modems.

```
10 REM   HEX DUMP ROUTINE
20 REM    FOR THE VZ-200
30 REM WRITTEN BY D.TESTER
40 REM     29/JUL/1984
50 REM
100 PRINT:PRINT" ADDRESS";
110 INPUT A$:GOSUB 3000
120 FOR X=1 TO 8:PRINT:GOSUB2000
125 PRINT" ";
130 FOR Y=1 TO 8
150 DA=PEEK(ADD):ADD=ADD+1:GOSUB1000:PRI
NT" ";
160 IFADD>32767THENADD=ADD-65536
180 NEXTY:NEXTX
190 PRINT:GOTO 100
999 END
1000 HEX=DA:' PRINTS SINGLE BYTE
1010 D1=INT(HEX/16):GOSUB 1030
1020 D1=HEX-INT(HEX/16)*16
1030 IF D1>9 THEN PRINTCHR$(D1+55); ELSE
 PRINTCHR$(D1+48);
1040 RETURN
1050 REM
2000 REM PRINTS DOUBLE BYTE
2010 IFSGN(ADD)=-1 THEN A1=ADD+65536 ELS
E A1=ADD
2020 HEX=INT(A1/256):GOSUB 1010
2030 HEX=A1-INT(A1/256)*256:GOSUB 1010
2040 RETURN
2050 REM
3000 REM CONVERTS HEX IN A$
3001 REM TO DECIMAL IN ADD
3010 IF LEN(A$)=0 THEN RETURN
3020 ADD=0:FOR Z=1 TO LEN(A$)
3030 H1=ASC(MID$(A$,Z,1))
3040 H1=H1-48:IFH1>9THENH1=H1-7
3050 ADD=ADD*16+H1:NEXTZ
3060 IF ADD>32767 THEN ADD=ADD- 65536
3070 RETURN
```

VZ-TRICKS by R. Constantine

PRINT

When entering a program line or in command mode it is possible to substitute a question mark "?" for the word "PRINT".

```
eg.  10 ? "ABS"
     10 PRINT "ABC"
```

When you list this line you will find the "?" has been replaced by the word "PRINT".

Try 10 ?A:?B:?C:?D:?E:?F:?G:?H:?I:?J:?K:?L:?M:?N:?O:?P:?Q:?R:?S:?T:?U

LIST and you get

```
10 PRINTA:PRINTB:PRINTC:PRINTD:PRINTE:PRINTF:PRINTG:PRINTH:P
RINTI:PRINTJ:PRINTK:PRINTL:PRINTM:PRINTN:PRINTO:PRINTP:PRINT
Q:PRINTR:PRINTS:PRINTT:PRINTU
```

Try fitting that on a line any other way!!
DANGER if you edit this line only the first two lines will remain, the rest is lost.

REM

When entering a REMark line or statement substitute an apostrophy "'" for the word "REM".

```
eg. 20 REM ENTRY POINT
    20 ' ENTRY POINT

eg. 30 SC=SC+1:REM ADDS ONE TO SCORE
    30 SC=SC+1:'ADDS ONE TO SCORE
```

THEN

When using "IF - THEN" statements substitute a comma "," for the word "THEN".

```
eg. 40 IF X=1 THEN Y=Y+1
    40 IF X=1 , Y=Y+1
```

OTHER TIPS

from J.P.Leon, S.Derecent, D.Boyce and M.Downes.

```
     TRON   POKE 31003,1
     TROFF  POKE 31003,0
Beep on     POKE 30779,1
Beep off    POKE 30779,0
```

Free memory and free string space

```
POKE 30862,212 : POKE 30863,39
PRINT USR(X) ; USR(X$)
```

The DELETE command (token 182) mentioned last issue can be used in the following formats,

```
          DELETE n      delete line n
          DELETE -n     delete from start to line n
  * —     DELETE n-     delete line n to end of program
          DELETE n-m    delete lines n to m
```

* Does not work without ordered line no

ON ERROR GOTO

As a result of my comment about the RESUME command last month two readers have replied with methods for getting into an error handling routine.
From C. Hudson, if A is the first line number of the error handling routine then

ON ERROR GOTO A is implemented by the following

POKE 30961,INT(A/256) : POKE 30960,A-256*INT(A/256)

However 30960 and 30961 are set to zero when RUN is used so the POKE's should be done from within the BASIC program.

From S. Debreceny two practical examples of his method for entering an error handling routine. The following is quoted from his letter.

Now, I would like to show you how I have been using error traps in my programming, and have included some listings of extracts from programs that you can enter and see working.

Error trapping of some errors can be very helpful in exchange of data between user and computer and also can speed up your MODE(0) and MODE(1) graphics programs enormously. The two listings I have enclosed are examples of both of these facilities. Firstly, I'll deal with user/computer data exchange with an example of the division by zero error trap.

Whilst running a program, a user may encounter a request for numeric data (ie INPUT A). If a mathematical function needs to be applied to variable A, it may be found that a value of A has been entered that could cause a divide by zero subsequently a DIVIDE BY ZERO error, which halts the program, will be displayed leaving the user bewildered as to what he has said to upset the microprocessor.

So, instead, try trapping the error, go back to the input and give the mug, sorry, user a second chance. To do this we need to look at our 'invisible commands' We can see from the table (see last issue, ED) that the TRS 80 token for ERROR is 158. The VZ 200 token 158 is SOUND.

You must take my word for this, that indeed when token 158 stands alone it is interpreted as SOUND. However if token 158 is preceeded by token 161, which is ON it is interpreted as ERROR.

eg. line 10 158 20,2 is SOUND 20,2
line 10 161 158 141 100 is ON ERROR GOTO 100

The divide by zero error trap example listing I have included picks a random number between 0 and 10. It then divides 100 by this number and prints the function on the screen. Ultimately 0 will be selected as a random number, and as the computer attempts to divide 100 by 0, generates a DIVIDE BY ZERO error. This error is trapped, the error type and line in which the error occurred is printed and the program resumed at the line which selects a new random number.

Now for the ILLEGAL FUNCTION trap. This may take a little more explanation as to why it is employed, but I'm sure you will see how it will speed up your graphic programs considerably. The program listing is an extract from a game I have written. It is a MODE(1) graphics game and the section we are looking at involves an object that moves across the sceen shooting bullets randomly in all directions. First lets look at how we could do this without the error trap.

The program would randomly select a direction, it would then SET a point in that direction, check if off screen, SET the next point and RESET the first, check if off screen etc etc. The off screen check every time a point is to be SET is unnecessary and wastes lots of valuable microsecond in the attempt to approach real time graphics simulation.

By using the ILLEGAL FUNCTION error trap, the bullet is permitted to advance as it will until the stage is reached when SET command can not be executed because of an off screen co-ordinate. An ILLEGAL FUNCTION error occures, the error is trapped and the co-ordinates are reset for a new bullet. This allows those microseconds to be used for collision checking with other objects.

BASIC Program to demonstrate error trap on DIVIDE BY ZERO error

```
10 CLS
12 ON ERROR GOTO 100
14 LETA=INT(RND(0)*10)
16 PRINT"RANDOM NUMBER = 100/";A;"=";INT(100/A)
18 SOUND20,1:0,8:GOTO14
100 PRINT:PRINT:PRINT"*** E R R O R   T R A P ***"
102 PRINT" ERROR TYPE ";ERR/2+1
104 PRINT" IN LINE NO. ";ERL
106 PRINT" A=";A:SOUND30,1;0,8
108 PRINT:RESUME 14
```

Above program dissassembled

```
LINE NO. 10      132

LINE NO. 12      161  158  141 100

LINE NO. 14      140 A 213  216 ( 222 (0) 207 10)

LINE NO. 16      178 "RANDOM NUMBER = 100/";A;"="; 216 (100 208 A)

LINE NO. 18      158 20,1;0,8: 141 14

LINE NO. 100     178 : 178 : 178 "*** E R R O R   T R A P ***"

LINE NO. 102     178 " ERROR TYPE "; 195  209 2 205 1

LINE NO. 104     178 " IN LINE NO. "; 194

LINE NO. 106     178 " A=";A: 158 30,1;0,8

LINE NO. 108     178 : 159 14
```

Basic Program Dissassembly Utility

```
1000 NL=31465
1002 LPRINT
1004 LPRINT"LINE NO.";PEEK(NL+2)+256*PEEK(NL+3);TAB(16);
1006 FORI=NL+4TONL+255
1008 IFPEEK(I)=0THENLPRINT:GOTO1016
1010 IFPEEK(I)<32ORPEEK(I)>90THENLPRINTPEEK(I);:GOTO1014
1012 LPRINTCHR$(PEEK(I));
1014 NEXT
1016 NL=PEEK(NL)+PEEK(NL+1)*256
1018 K$=INKEY$:K$=INKEY$:IFK$=""THEN1018ELSESOUND20,1:GOTO1002
```

BASIC Program to demonstrate error trap on ILLEGAL FUNCTION error

```
10 MODE(1):COLOR2
12 ON ERROR GO TO 100
14 SX=4:SY=60:SET(SX,SY)
16 X=SX:Y=SY
18 A=RND(0)*2-1:IFABS(A)<.05THEN18
20 B=RND(0)*-1.8:IFABS(B)<.5THEN20
22 X=X+A:Y=Y+B
24 SET(X+A,Y+B)
26 RESET(X,Y)
28 SX=SX+.1:SET(SX,SY):RESET(SX-1,SY):GOTO22
100 IFSX>127THENRESET(SX-1,SY):RESET(X+A,Y+B):RESUME 14
102 RESET(X,Y):RESUME 16
```

Above program dissassembled

```
LINE NO.  10     157 (1): 151 2

LINE NO.  12     161  158  141 100

LINE NO.  14     SX 213 4:SY 213 60: 131 (SX,SY)

LINE NO.  16     X 213 SX:Y 213 SY

LINE NO.  18     A 213  222 (0) 207 2 206 1: 143  217 (A) 214 .05 202 18

LINE NO.  20     B 213  222 (0) 207  206 1.8: 143  217 (B) 214 .5 202 20

LINE NO.  22     X 213 X 205 A:Y 213 Y 205 B

LINE NO.  24      131 (X 205 A,Y 205 B)

LINE NO.  26      130 (X,Y)

LINE NO.  28     SX 213 SX 205 .1: 131 (SX,SY): 130 (SX 206 1,SY): 141 22

LINE NO.  100     143 SX 212 127 202  130 (SX 206 1,SY): 130 (X 205 A,Y 205 B)
59 14

LINE NO.  102     130 (X,Y): 159 16
```

```
2000 'BYTE TO BINARY CONVERSION
2010 'ON CALL BY=VALUE (<256)
2020 'ON RETURN A$=BINARY VALUE
2030 'USES BI BV RC
2040 'BY IS UNCHANGED
2045 A$=""
2050 FOR BI=7 TO 0 STEP -1
2060 BV=2^BI : RC=BY AND BV
2070 IF RC>0 THEN A$=A$+"1"ELSEA$=A$+"0"
2080 NEXT BI : RETURN
```

```
3000 'DECIMAL TO HEX CONVERSION
3010 'ON CALL BY=VALUE (<256)
3020 'ON RETURN A$=HEX VALUE
3030 'USES H1 H2
3040 'BY IS UNCHANGED
3045 TA$="0123456789ABCDEF":A$=""
3050 H1=INT(BY/16)+1
3060 H2=BY-16*(H1-1)+1
3070 A$=MID$(TA$,H1,1)+MID$(TA$,H2,1)
3080 RETURN
```

```
4000 'DECIMAL TO OCTAL COVERSION
4010 'ON CALL BY=VALUE (<256)
4020 'ON RETURN A$=OCTAL VALUE
4030 'USES O1 O2 O3
4040 'BY IS UNCHANGED
4050 TA$="0123456789ABCDEF":A$=""
4060 O1=INT(BY/64)
4070 O2=INT((BY-64*O1)/8)
4080 O3=BY-64*O1-8*O2
4085 O1=O1+1:O2=O2+1:O3=O3+1
4090 A$=MID$(TA$,O1,1)+MID$(TA$,O2,1)+MI
D$(TA$,O3,1)
4100 RETURN
```

ANY BASE TO DECIMAL

```
5000 'ON CALL A$=NUMBER
5010 '        PO=BASE EG 2 FOR BINARY
5020 'ON RETURN BY=DECIMAL VALUE
5030 'USES DI NC MU DV
5040 'A$ AND PO ARE UNCHANGED
5050 NC=LEN(A$) : BY=0
5060 FOR DI= 1 TO NC
5070 MU=PO^(NC-DI)
5080 DV=ASC(MID$(A$,DI,1))
5090 IFDV>57THENDV=DV-55ELSEDV=DV-48
5100 BY=BY+DV*MU
5110 NEXT DI
5120 RETURN
```

EXAMPLE OF A PROGRAM TO USE THE ROUTINES

```
10 INPUT"DECIMAL VALUE";BY
20 IFBY>255THENPRINT"TOO BIG":GOTO10
30 GOSUB 2000
40 PRINT"DECIMAL";BY;"IS BINARY ";A$
50 GOTO 10
```

THESE CHANGES GIVE DECIMAL TO HEX

```
30 GOSUB 3000
40 PRINT"DECIMAL";BY;"IS HEX ";A$
```

THESE CHANGES GIVE DECIMAL TO OCTAL

```
30 GOSUB 4000
40 PRINT"DECIMAL";BY;"IS OCTAL ";A$
```

```
5 'HEX TO BINARY AND DECIMAL
10 INPUT"HEX NUMBER";A$
20 PO=16
30 GOSUB5000
40 H$=A$
50 GOSUB2000
60 PRINT"HEX ";H$;" IS"
70 PRINT"BINARY ";A$;" AND"
80 PRINT"DECIMAL ";BY
90 GOTO10
```

LOTTO SELECTOR

This program uses the same line numbers as the utility listed last month. This makes it difficult to use the utility to implement the 'RANDOM' function in line 8. However, since there is only 1 invisible command to be implemented it can be done in the immediate mode. Do this as follows,

Enter line 8 as the first line thus

8 CLS:REM

then enter this command

POKE 31471,134

Line 8 will now clear the screen and randomize the RND function. Enter the rest as normal.

```
5 REM LOTTO SELECTOR
6 REM COPYRIGHT J. WATERS
7 REM   AUGUST 1984
8 CLS:RANDOM
10 PRINT@9,"LOTTO SELECTOR"
15 PRINT@67,"YOU CAN SELECT FOR"
20 PRINT@135,"1 X-LOTTO"
30 PRINT@167,"2 MID WEEK LOTTO"
35 PRINT@199,"3 6 FROM 36 POOLS"
40 PRINT@259,"ENTER YOUR CHOICE";
45 INPUTJ
47 IF J<1 OR J>3 THEN 40
50 PRINT@355,"HOW MANY NUMBERS DO"
55 PRINT@387,"YOU WANT";
60 INPUTL
70 IF J=2 THEN MN=39 ELSE MN=40
75 IF J=3 THEN JJ=5:MN=36 ELSE JJ=6
105 REM SET UP DISPLAY:CLS
110 CLS
115 DATA96,113,96,114,96,115,96,116,96,1
17,96,118,96,119
120 DATA96,120,96,121,113,112,113,113,11
3,114,113,115,113,116
125 DATA113,117,113,118,113,119,113,120,
113,121,114,112,114,113
130 DATA114,114,114,115,114,116,114,117,
114,118,114,119,114,120
135 DATA114,121,115,112,115,113,115,114,
115,115,115,116,115,117
140 DATA115,118,115,119,115,120,115,121,
116,112,96,96,96,96
150 J=MN
155 CLS
156 FORK=0TO5
160 N=32*K+28773
165 FORKK=0TOJJ
167 IF J=0 THEN 185
170 NN=N+3*KK
175 READA1,A2
180 POKENN,A1:POKENN+1,A2
182 J=J-1
185 NEXTKK:NEXTK
190 RESTORE
205 REM SELECT & DISPLAY
207 PRINT@0,"YOUR NUMBERS ARE";
210 FORK=1TOL
215 NUX=RND(MN):IX=(NUX-1)/(JJ+1)
220 IIX=NUX-IX*(JJ+1)
225 PO=28770+3*IIX+32*IX
230 A1=PEEK(PO)
235 IFA1<64THEN215ELSEA1=A1-64
240 A2=PEEK(PO+1)-64
245 POKEPO,A1:POKEPO+1,A2
250 NEXTK
305 REM DISPLAY NUMBERS IN ORDER
310 PO=29029
315 FORK=0TO5
320 N=32*K+28773
325 FORKK=0TOJJ
330 NN=N+3*KK
335 IFPEEK(NN)>64THEN355
340 A1=PEEK(NN)+64:A2=64+PEEK(NN+1)
345 POKEPO,A1:POKEPO+1,A2
350 IFPO=29044THENPO=29061ELSEPO=PO
355 NEXTKK:NEXTK
360 PRINT@448,"PRESS N FOR NEXT SELE
N";
362 PRINT@480,"      M FOR MENU
 ";
365 Q$=INKEY$
370 IFINKEY$="N"THEN150
375 IFINKEY$="M"THEN8ELSE365
380 END
```

EXAMPLE OF DISPLAYS

```
        LOTTO SELECTOR

   YOU CAN SELECT FOR

       1 X-LOTTO
       2 MID WEEK LOTTO
       3 6 FROM 36 POOLS

   ENTER YOUR CHOICE? 1


   HOW MANY NUMBERS DO
   YOU WANT? 6
```

```
YOUR NUMBERS ARE

     [1]  2  3  4  5  6  7
      8  9 [10] 11 12 [13] 14
     15 16 [17] 18 [19] 20 21
     22 23 24 25 26 27 28
     29 [30] 31 32 33 34 35
     36 37 38 39 40


      1 10 13 17 19 30


PRESS N FOR NEXT SELECTION
      M FOR MENU
```

BASIC MONITOR FOR THE VZ200 BY R G DAVIS

ALTHOUGH THE TECHNICAL MANUAL FOR THE VZ 200 CONTAINS A LOT OF GOOD INFORMATION THERE ARE A FEW IMPORTANT THINGS IT DOES NOT TELL YOU.THE ONE WE WILL LOOK AT HERE IS THAT ALTHOUGH THEY TELL YOU HOW TO RESERVE SPACE FOR A MACHINE CODE PROGRAM AND HOW TO CALL THEM, THEY HAVE NOT TOLD YOU THAT TO POKE TO ADDRESSES ABOVE 32767 YOU HAVE TO USE TWO'S COMPLEMENT.

AS AN EXAMPLE IF YOU HAVE RESERVED 255 BYTES AT THE TOP OF YOUR 8K MEMORY FROM 36608 TO 36863(8F00 - 8FFF) INSTEAD OF POKE 36608,X YOU WILL HAVE TO USE POKE-28928,X
TO FIND THE TWO'S COMPLEMENT OF THE ADDRESS YOU REQUIRE USE THE FOLLOWING FORMULA -1*(65536-DESIRED ADDRESS).

ALSO WHEN YOU WRITE YOUR MACHINE CODE PROGRAMS YOU THEN HAVE TO CONVERT IT INTO DECIMAL TO POKE INTO MEMORY AND WHEN YOU PEEK THE PRINT OUT IS ALSO IN DECIMAL.

WELL THIS PROGRAM GETS AROUND ALL OF THESE PROBLEMS FOR YOU WHEN YOU RUN IT YOU GET A CHOICE OF THREE FUNCTIONS, YOU CAN STEP THROUGH MEMORY ONE BYTE AT A TIME AND ENTER NEW DATA IN HEX IF YOU WISH OR PRESS ENTER AND STEP ON TO THE NEXT BYTE.

THE NEXT CHOICE IS THAT YOU CAN VIEW ON THE SCREEN 16 LINES BY 8 COLUMNS (128 BYTES)OF HEX DATA AT A TIME TO VERIFY THE DATA YOU HAVE ENTERED THEN PRESS SPACE TO BRING UP THE NEXT 16 LINES ETC.

THE LAST OPTION IS THAT YOU CAN THEN USE YOUR PRINTER TO GET A COPY OF THE PROGRAM IN HEX WITH 16 BYTES ON EACH LINE, OR IF YOU WANT TO CHANGE THIS TO 8 BYTES THEN CHANGE THE END OF LINE 9 TO:FOR N=1 TO 8

THE FOLLOWING IS AN EXAMPLE OF THE PRINT OUT GIVING YOU THE FIRST FEW LINES OF THE BASIC ROM.....

```
0000  F3 AF 32 00 68 C3 74 06
0008  C3 00 78 E1 E9 00 00 00
0010  C3 03 78 C5 06 01 18 2E
0018  C3 06 78 C5 06 02 18 26
0020  C3 09 78 C5 06 04 18 1E
0028  C3 0C 78 11 15 78 18 E3
0030  C3 0F 78 11 1D 78 18 E3
0038  C3 B8 2E 11 25 78 18 DB
0040  C3 FD 2E C9 00 00 C3 C2
0048  03 CD 2B 00 B7 C0 18 F9
0050  2A 20 78 7E 32 3C 78 C9
0058  4C FE 54 20 D6 FD 21 F1
```

```
0000  F3 AF 32 00 68 C3 74 06 C3 00 78 E1 E9 00 00 00
0010  C3 03 78 C5 06 01 18 2E C3 06 78 C5 06 02 18 26
0020  C3 09 78 C5 06 04 18 1E C3 0C 78 11 15 78 18 E3
0030  C3 0F 78 11 1D 78 18 E3 C3 B8 2E 11 25 78 18 DB
0040  C3 FD 2E C9 00 00 C3 C2 03 CD 2B 00 B7 C0 18 F9
0050  2A 20 78 7E 32 3C 78 C9 4C FE 54 20 D6 FD 21 F1
0060  0B 78 B1 20 FB C9 31 00 06 3A EC 68 3C FE 02 D2
0070  00 00 C3 CC 06 11 80 78 21 F7 18 01 27 00 ED B0
```

NOTE, ALL ' SHOULD BE ENTERED AS "

```
1 CLS
2 PRINT'MEMOD       M'
3 PRINT'REVIEW      R'
4 PRINT'PRINTOUT    P'
5 INPUT Y$
6 IF Y$='M' THEN 12
7 IF Y$='R' THEN 17
8 GOSUB 30
9 GOSUB 61:LPRINT H$'  ';:FOR N=1 TO 16
10 GOSUB 36:LPRINT H$' ';
11 GOSUB 38:NEXT:LPRINT:GOTO 9
12 GOSUB 30:CLS
13 GOSUB 61:PRINT H$'  ':GOSUB 36
14 PRINT H$'  ';:INPUT B$
15 IF B$='' THEN 16 ELSE GOSUB 42:POKE M,F
16 GOSUB 38:GOTO 13
17 GOSUB 30:CLS
18 PRINT:FOR T=1 TO 16
19 PRINT:GOSUB 61:PRINT H$'  ';:FOR N=1 TO 8
20 GOSUB 36:PRINT H$' ';
21 GOSUB 38:NEXT:NEXT
22 Z$=INKEY$:W$=INKEY$:IF W$<>' ' THEN 22
23 GOTO 18
30 INPUT'START ADDRESS';M:B=M
31 IF M<32768 THEN 33 ELSE Q=M
32 GOSUB 40:M=X
33 INPUT'END ADDRESS';E
34 IF E<32768 THEN RETURN ELSE Q=E
35 GOSUB 40:E=X:RETURN
36 D=PEEK(M):DE=D:IF D<32 THEN DE=46
37 H$='':GOSUB 65:RETURN
38 M=M+1:IF M=32768 THEN GOSUB 41
39 IF M=E THEN END ELSE B=B+1:GOTO 61
40 X=-1*(65536-Q):RETURN
41 Q=M:GOSUB 40:M=X:RETURN
42 C$=B$:F=0:GOSUB 48
43 RETURN
44 IF C<58 THEN C=C-48
45 IF C>63 THEN C=C-55
46 RETURN
48 C=ASC(C$)
49 GOSUB 44:G=16*C:F=F+G
50 C$=RIGHT$(B$,1):C=ASC(C$)
51 GOSUB 44:F=F+C:RETURN
61 H$='':A=B/4096:A=INT(A):GOSUB 67
62 D=B-4096*A
63 A=D/256:A=INT(A):GOSUB 67
64 D=D-256*A
65 A=D/16:A=INT(A):GOSUB 67
66 A=D-16*A
67 AB=A:IF A>9 THEN A=A+55 ELSE A=A+48
68 A$=CHR$(A):H$=H$+A$:A=AB:RETURN
```

Line 9 - Gosub 61 . . . . . :For N=1 To 24

```
5 REM FLASH-WORDS PROGRAMME
10 REM 398 WORDS
20 DIMA$(398)
30 CLS
31 FORJ=1TO15
32 PRINT"FLASH - WORDS                    ";
33 NEXT
34 PRINT"FLASH - WORDS";
35 CLS
40 FORJ=1TO15
60 PRINT"FLASH - WORDS                    ";
65 SOUND30,2:SOUND20,4
70 NEXT
80 PRINT"FLASH - WORDS";
81 CLS
82 FORJ=1TO15
83 PRINT"FLASH - WORDS                    ";
84 NEXT
85 PRINT"FLASH - WORDS";
90 FORJ=1TO5000:NEXT
1000 FORX=1TO398
1100 READ  A$(X)
1200 NEXT
1400 D=0:SC=0:ER=0:NW=0:CW=0
1410 R$(1)="PERFECT SCORE"
1420 R$(2)="EXCELLENT"
1430 R$(3)="VERY GOOD"
1440 R$(4)="GOOD"
1450 R$(5)="POOR"
1460 R$(6)="KEEP TRYING"
1470 R$(7)="TRY LOWER LEVEL"
1480 R$(8)="DO YOU SPEAK FRENCH?"
1500 CLS
1600 INPUT"DIFFICULTY 1/2/3/4/5/6";DF
1610 IFDF>6ORDF<1GOTO1600
1620 IFDF=1THEND=1:DD=1
1630 IFDF=2THEND=1:DD=2
1640 IFDF=3THEND=2:DD=2
1650 IFDF=4THEND=2:DD=3
1660 IFDF=5THEND=3:DD=3
1670 IFDF=6THEND=3:DD=4
1700 INPUT"NO. OF WORDS";NW
2000 CLS
2010 PRINT@490,"SCORE = ";SC
2040 IFCW>=NWGOTO3000
2050 B$=A$(RND(398))
2055 L=LEN(B$)
2060 IFD=1ANDL>4GOTO2050
2065 IFD=2AND(L>8ORL<3)GOTO2050
2070 IFD=3ANDL<4GOTO2050
2100 PRINT@203,B$
2300 FORW=1TO900/DD:NEXT
2350 CLS
2360 PRINT@490,"SCORE = ";SC
2500 PRINT@10,"":INPUTZ$
2600 IFZ$=B$THENSC=SC+1:CW=CW+1:SOUND30,1:SOUND30,1:GOTO2000
2650 ER=ER+1
2700 SOUND10,7: GOTO2100
3000 REM FINAL SCORES
3100 CLS
3200 PRINT"          FLASH-WORDS":PRINT
3300 PRINT"  NO. WORDS = ";NW
3400 PRINT
3500 PRINT " SCORE = ";SC
3600 PRINT
3700 PRINT"  ERRORS = ";ER
3800 PRINT
3850 P=((SC-ER)/SC)*100
3900 PRINT"  PERCENTAGE = ";P;"%"
3910 PRINT
3920 IFP=100THENR=1
3921 IFP<100ANDP=>90THENR=2
3922 IFP<90ANDP=>70THENR=3
3923 IFP<70ANDP=>50THENR=4
3924 IFP<50ANDP=>40THENR=5
3925 IFP<40ANDP=>20THENR=6
3926 IFP<20ANDP=>0THENR=7
3927 IFP<0THENR=8
3980 PRINT"  RATING = ";R$(R)
4000 PRINT:PRINT:PRINT:PRINT
4100 INPUT"MORE WORDS Y/N";XX$
4200 IFXX$="Y"GOTO1400
4300 END
```

Flash-Words

This program was written by Ross Constantine and is described by him as follows.

"This program flashes on the screen a word which the player must remember then type in correctly before they progress to the next word.

The player has a choice of difficulty and number of words. The degree of difficulty varies the length of time the word is displayed on the screen and the length of word displayed. Readers can add extra words by adding additional DATA statements or adding words to existing DATA statements. Lines 20, 1000 & 2050 will need to be changed to the new number of words."

```
30200 DATASUMMER,OUR,AN,WHAT,MANY,BEST,CRY
30600 DATABUTTER,CHURCH,DOES,TEACHER,CHILDREN
30700 DATAMADE,GAVE,MATE,BROTHER,DISCOVER
30800 DATASISTER,INSIDE,MORNING,AUTUMN,STORE
30900 DATANINE,SMILE,HIDE,TENT,QUICK,WEEK,BUY
31000 DATANEWS,PRETTY,STORY,LIBRARY,VISIT,EXPLORE
31200 DATAHOLE,BONE,SMOKE,WASH
31300 DATADIRTY,DINNER,AFTERNOON,WANT,WEEKEND,SATURDAY
31600 DATACORNER,FRONT,ANYONE,ANYTHING,SUNDAY,MONDAY,APRIL
31700 DATAEVENTS,WATCHED,FAMILY,CARAVAN
31800 DATAOFF,KING,GOING,CHICKEN,GONE
31900 DATAHAPPY,BECAUSE,FARMER,CALF,BESIDE,RABBIT,EASTER,OUTING
32100 DATANUT,TALL,LOAF,BUCKET
32200 DATAWATER,WERE,SOMETHING,MAY,MONTH,EVERY,FRIDAY
32300 DATAKNOW,BORROW,FAVOURITE,TELEVISION
32400 DATAROPE,ROSE,LIVE,SHOW
32500 DATATHERE,FOUND,MOUSE,PUSH,PICNIC,COAST
32600 DATATRAVEL,ENJOY,RAILWAY,ENGINE,CANBERRA
40300 DATAFUNNY,BE,LOOK,TWO,IT,SO,HIM,EYE,IN,BIG,A
40700 DATASIT,IS,HE,MILK,ON,COT,DOG,DOLL
40800 DATAGIRL,AT,RAN,CAT,FOR
40900 DATAWHEN,GET,RED,HEN,HOP,BIRD
41000 DATARUN,BUS,MUM,FUN
41100 DATAYELLOW,MAN,JET,DAD,HUG,FLY,FLAG
41200 DATABLUE,OF,OVER,GO,TAP,TREE
41300 DATATHEY,HOOK,LINE,CATCH,BOAT,PULL,FISHING,I,AM,JAM,SIX
41400 DATACUP,TELL,BELL,ARE,PARK,GARDEN,SWING,SEE-SAW
41500 DATAFLOWER,GRASS,TO,DO,RUG,MUD,YES,HAS,WILL
41600 DATAFARM,YARD,KITTEN,BROWN,KIND,FOOL,HORSE,VAN,TOP
41700 DATATHE,PRAM,BACK,PLAY,MEET,GREEN,COME,TODAY,CLOCK
41800 DATATIME,BEGIN,AGO,START,AFTER,US,LET,SUN,ME,FROG
41900 DATADUCK,LITTLE,FISH,SHIP,SOME,STOP,STREET,STAND,STILL
42000 DATACROSS,ROAD,WALK,ACROSS,RULES,SAFETY,TEN,PIG,HAD,MY
42100 DATAGRUB,PICK,THIS,THEM,THAT,WAS,SEND,CARD,STAMP,POST
42200 DATAWRITE,LETTER,READ,NAME,PAPER,MAIL,HAT,BAD,PIN,SEE,DRUM
42300 DATANECK,SCHOOL,ROOM,FOOD,PUT,THIN,CAR,BIKE,RIDE,THICK
42400 DATASMALL,SEAT,SHORT,LONG,VERY,UP,FOX,CAN,BOY,GLAD,SOCK
42500 DATAWITH,HER,MOTHER,FATHER,MOON,STAR,SKY,DARK,TORCH,BLACK
42600 DATASHINE,NIGHT,SPACE,ROCKET,FIX,LED,MUG,AND,POND
42700 DATAHOME,COW,DOWN,YOU,SHOP,EGG,EAT,LUNCH,MEAT,TEA,DRINK
42800 DATACAKE,APPLE,ORANGE,HUT,NOT,PET,WE,JUMP,CAMP,HAVE,BOOK
42900 DATAGOOD,SAID,DRAW,DESK,MAKE,PAINT,TABLE,CHAIR,PASTE
43000 DATAPLAYTIME,CLASS,HAM,TIN,MOP,INTO,ANT,WENT,BABY,ARM
43100 DATASAW,HAND,FEET,TEETH,BRUSH,CHEEK,NOSE,BIGGER,GROW,FACE
43200 DATABODY,HIS,IF,DI  BY,MUST,NEST,LIKE,SONG,RING,ALL
43300 DATAOUT,MYSELF,THOSE,YOUR,GIVE,ONE,FIRST,ANY,FRIEND,HIT
43400 DATALEG,BOX,FROM,SHE,SWIM,BALL,BOOT,THAN,PINK,JUMPER
43500 DATASOCKS,COTTON,WOOL,SHIRT,SKIRT,JEANS,OLD,NEW,WHO,AWAY
43600 DATANO,IT'S,THREE,FIVE,SOON,UNDER,CALL,FAST,FIND,HELP
43700 DATADON'T,CARRY,COLD,CAME,HILL,WOOD,SHOE,TOYS,BREAD,TOO
43800 DATASHEEP,BEAR,PONY,PUPPY
```

I've had quite a few requests for information on machine code programming. I am working on a couple of "introduction to MCr" type articles so perhaps some of you who have already mastered the art (sometimes I think its a black art) could help me out here.